Interview Summary	Application No.	Applicant(s)	
	10/083,891	BAKER ET AL.	
	Examiner	Art Unit	
	Salad E. Abdullahi	2157	

All participants (applicant, applicant's representative, PTO personnel):

(1) *Salad E. Abdullahi*. (3)_____.

(2) *Brian D. Kirkpatrick*. (4)_____.

Date of Interview: *04 February 2008*.

Type: a)☒ Telephonic b)☐ Video Conference
c)☐ Personal [copy given to: 1)☐ applicant 2)☐ applicant's representative]

Exhibit shown or demonstration conducted: d)☐ Yes e)☐ No.
If Yes, brief description: _____.

Claim(s) discussed: *8, 9 and 16*.

Identification of prior art discussed: _____.

Agreement with respect to the claims f)☒ was reached. g)☐ was not reached. h)☐ N/A.

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: *Examiner proposed claims 8, 9 and 16 be amended to remove certain claim language that lacks support from the specification. Applican agreed*.

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims allowable, if available, must be attached. Also, where no copy of the amendments that would render the claims allowable is available, a summary thereof must be attached.)

THE FORMAL WRITTEN REPLY TO THE LAST OFFICE ACTION MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a reply to the last Office action has already been filed, APPLICANT IS GIVEN A NON-EXTENDABLE PERIOD OF THE LONGER OF ONE MONTH OR THIRTY DAYS FROM THIS INTERVIEW DATE, OR THE MAILING DATE OF THIS INTERVIEW SUMMARY FORM, WHICHEVER IS LATER, TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. See Summary of Record of Interview requirements on reverse side or on attached sheet.

Examiner Note: You must sign this form unless it is an Attachment to a signed Office action.

_____
Examiner's signature, if required

1. (Canceled)

2. (Previously presented) A method of tunneling a transaction based protocol through a generic Internet protocol (IP) transport, the method comprising:

providing a generic messaging structure that includes a transport protocol, a message buffer, a source-address field and one or more data fields for transparent routing of a user protocol over the IP transport during a host-to-host communication or telecommunication session;

providing an application program interface to the generic messaging structure, the application program interface including a mechanism for a user to choose a desired transport and associated protocol for transparently routing the user protocol over the desired transport in accordance with the chosen transport protocol within the one or more data fields;

creating a base class library including plural defined source and header files, and

providing a mechanism for deriving a transaction-based protocol-specific class that is compatible with the base class library, the transaction-based protocol-specific class further being derived based in part on the chosen transport protocol.

3. (Previously presented) The method of claim 2, wherein the transaction-based protocol-specific class is derived using an object-oriented inheritance based mechanism.

4. (Previously presented) The method of claim 2 including compiling the transaction-based protocol-specific class when a transaction and the transport protocol are determined.

5. (Previously presented) A method of tunneling any related data-, control-, or routing-related protocol through a generic Internet protocol (IP) transport, the method comprising:

creating a base class library including plural defined source and header files, the base class library further including base class constructors of virtual, copy, and assignment, and generic access methods;

choosing a transport protocol for transparently routing a user protocol over the IP transport; and

providing a mechanism for deriving a transaction-based protocol-specific class that is compatible with the base class library, the transaction-based protocol-specific class further being derived based in part on the chosen transport protocol, wherein the transaction-based protocol-specific class is derived using an object-oriented inheritance based mechanism.

6. (Previously presented) The method of claim 5 which further comprises:

providing a generic messaging structure that includes the transport protocol, a message buffer, a source-address field and one or more data fields for transparent routing of the user protocol over the IP transport during a host-to-host communication or telecommunication session.

7. (Previously presented) The method of claim 5 which further comprises:

providing an application program interface to a generic messaging structure, the interface including a mechanism for a user to choose a desired transport and associated protocol for transparently routing the user protocol over the transport in accordance with the chosen transport protocol within the generic messaging structure.

8. (Currently amended) A method of tunneling any related data-, control-, or routing-related protocol through a generic Internet protocol (IP) transport, the method comprising:

creating a base class library including plural defined source and header files, the base class library further including base class constructors of virtual, copy, and assignment, and generic access methods;

providing a mechanism for deriving a transaction-based protocol-specific class that is compatible with the base class library, the transaction-based protocol-specific class further being derived based in part on the chosen transport protocol;

providing an application program interface to a generic messaging structure, the interface including a mechanism for a user to choose the transport protocol; and

provide a generic messaging structure that includes the transport protocol, a message buffer, a source-address field and one or more data fields for transparent routing of a user protocol over the IP transport during a host-to-host communication or telecommunication session.

~~compiling the transaction-based protocol-specific class prior to a run-time selection of the chosen transport protocol, and~~

~~selecting at run-time the pre-compiled transaction-based protocol-specific class for the chosen transport protocol, wherein the transaction-based protocol-specific class is derived using an object-oriented inheritance based mechanism~~.

9. (Currently amended) A computer-readable medium including an application programming interface stored on the computer-readable medium for transparently routing data between hosts in an Internet protocol (IP) transport, comprising:

a message buffer data structure defining a protocol-generic parent class, message, source-address and data fields for a chosen transport protocol;

a message creation mechanism for creating a message and adding it to the message buffer data structure; and

a protocol creation mechanism for deriving a protocol-specific child class based on the chosen transport protocol that renders new protocol-specific sub-fields of the protocol field of the message buffer data structure, where the application programming interface includes a mechanism for a user to choose a desired transport and associated protocol for transparently routing a user protocol over the transport in accordance with the chosen transport protocol within the generic messaging structure ~~is operable to:~~

~~compile the transaction-based protocol-specific class prior to a run-time selection of the chosen transport protocol, and~~

~~select at run-time the pre-compiled transaction-based protocol-specific class for the chosen transport protocol~~.

10. (Previously presented) The computer-readable medium of claim 9 in which the protocol-specific child class is derived using an object-oriented inheritance based mechanism.

11. (Cancelled)

12. (Previously presented) The computer-readable medium of claim 10, wherein the message creation and protocol creation mechanisms include computer-readable and computer-executable software instructions.

13. (Previously presented) The computer-readable medium of claim 12, including software source code and headers in C/C++ programming language form.

14. (Canceled)

15. (Previously presented) A computer-readable medium containing a program for tunneling a transaction based protocol through a generic Internet protocol (IP) transport, wherein when the program is executed by at least one device it is operable to:

provide a generic messaging structure that includes a transport protocol, a message buffer, a source-address field and one or more data fields for transparent routing of a user protocol over the IP transport during a host-to-host communication;

provide an application program interface to the generic messaging structure, the application program interface including a mechanism for a user to choose a desired transport and associated protocol for transparently routing the user protocol over the desired transport in accordance with the chosen transport protocol within the one or more data fields;

create a base class library including plural defined source and header files, and

provide a mechanism for deriving a transaction-based protocol-specific class that is compatible with the base class library, the transaction-based protocol-specific class further being derived based in part on the chosen transport protocol.

16. (Currently amended) A computer-readable medium containing a program for tunneling a data-related protocol through a generic Internet protocol (IP) transport, wherein when the program is executed by at least one device it is operable to:

create a base class library including plural defined source and header files, the base class library further including base class constructors of virtual, copy, and assignment, and generic access methods;

choose a transport protocol for transparently routing a user protocol over the IP transport; and

provide a mechanism for deriving a transaction-based protocol-specific class that is compatible with the base class library, the transaction-based protocol-specific class further being derived based in part on a chosen transport protocol, wherein the transaction-based protocol-specific class is derived using an object-oriented inheritance based mechanism;

~~compile the transaction-based protocol-specific class prior to a run-time selection of the chosen transport protocol; and~~

~~select at run-time the pre-compiled transaction-based protocol-specific class for the chosen transport protocol~~.

17. (Previously presented) The computer-readable medium in accordance with claim 16, wherein the program is further operable to:

provide an application program interface to a generic messaging structure, the interface including a mechanism for a user to choose a desired transport and associated protocol for transparently routing a user protocol over the transport in accordance with the chosen transport protocol within the generic messaging structure.

18. (Previously presented) The method of claim 2 wherein the transport protocol is operated on by a signaling function and wherein the user protocol may be routed over the transport without a switching function.

19. (Previously presented) The method of claim 2 including populating a message structure of the transaction-based protocol-specific class with tag-length-value (TLV) trios when the transaction-based protocol-specific class is derived.

20. (Previously presented) The computer-readable medium in accordance with claim 16, wherein the program is further operable to:

provide a generic messaging structure that includes the transport protocol, a message buffer, a source-address field and one or more data fields for transparent routing of a user protocol over the IP transport during a host-to-host communication or telecommunication session.